UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/737,374	12/16/2003	Roger Hansen	200312027-1	5369

22879 7590 03/09/2009

HEWLETT PACKARD COMPANY
P O BOX 272400, 3404 E. HARMONY ROAD
INTELLECTUAL PROPERTY ADMINISTRATION
FORT COLLINS, CO 80527-2400

EXAMINER	
TRUONG, LOAN	

ART UNIT	PAPER NUMBER
2114	

NOTIFICATION DATE	DELIVERY MODE
03/09/2009	ELECTRONIC

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

JERRY.SHORMA@HP.COM
mkraft@hp.com

Office Action Summary	Application No.	Applicant(s)	
	10/737,374	HANSEN ET AL.	
	Examiner	Art Unit	
	LOAN TRUONG	2114	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *15 December 2008*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-16 and 18-37* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-16 and 18-37* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

FINAL ACTION

1. This office action is in response to the Request for continuation filled December 15, 2008 in application 10/737,374.

2. Claims 1-16 and 18-37 are presented for examination. Claims 1, 10, 21, 26, 32 and 35 are amended. Claim 17 is previously cancelled.

Response to Arguments

3. Applicant's arguments with respect to claims 1-16 and 18-37 have been considered but are moot in view of the new ground(s) of rejection.

Continued Examination Under 37 CFR 1.114

4. A request for continued examination under 37 CFR 1.114, including the fee set forth in 37 CFR 1.17(e), was filed in this application after final rejection. Since this application is eligible for continued examination under 37 CFR 1.114, and the fee set forth in 37 CFR 1.17(e) has been timely paid, the finality of the previous Office action has been withdrawn pursuant to 37 CFR 1.114. Applicant's submission filed on December 15, 2008 has been entered.

Claim Objections

5. Claim 30 is objected to because of the following informalities: Claim 30 line 2 has "seets" which examiner interpreted as sets. Appropriate correction is required.

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

The factual inquiries set forth in *Graham* **v.** *John Deere Co.*, 383 U.S. 1, 148 USPQ 459 (1966), that are applied for establishing a background for determining obviousness under 35 U.S.C. 103(a) are summarized as follows:

1. Determining the scope and contents of the prior art.
2. Ascertaining the differences between the prior art and the claims at issue.
3. Resolving the level of ordinary skill in the pertinent art.
4. Considering objective evidence present in the application indicating obviousness or nonobviousness.

6. Claims 1-5, 7-13, 18-19, 21, 25-26, 28, 32-33 and 35-37 are rejected under 35 U.S.C. 103(a) as being unpatentable over Chung et al. (US 6,195,760) in further view of Beukema et al. (US 2002/0124117).

In regard to claim 1, Chung et al. teach a system for storing checkpoint state information, comprising:

a network interface to an external network (*network, fig. 1, 100, col. 3 lines 60-62)*; and

a persistent memory unit (*Checkpoint Server, fig. 1, 110, col. 4 lines 41-44)* coupled to the network interface (*connected to the network, fig. 1, 100, col. 4 lines 41-44*, wherein:

the persistent memory unit (*Checkpoint Server, fig. 1, 110, col. 4 lines 41-44)* is configured to receive the checkpoint data (*periodically receives from each fault-protected application module running on the network the most current state of that application, col.*

4 lines 41-44) from a primary process through the network interface (*hot backup where each copies of an application can process client request and states are synchronized among multiple copies, col. 2 lines 7-14)*, and to provide access the checkpoint data from the backup process (*last operating state provided to the backup, col. 4 lines 34-40)* through the network interface (*last stored state is retrieved from the memory of Checkpoint Server connected to network, fig. 1, 110, 100, col. 4 lines 41-48)*; and

the backup process provides recovery capability in the even of a failure of the primary process (*idle or backup application module assume the functioning of a failed primary application module upon failure-detection, col. 4 lines 34-40)*.

Chung et al. does not explicitly teach the system for storing checkpoint state information comprising a persistent memory unit receive data into a region of the persistent memory unit via a remote direct memory write command and to provide access to the data in the region via a remote direct memory read command, wherein the remote direct memory write command is preceded by a create request for the region and the read command is preceded by an open request for the region.

Beukema et al. teach a infiniband memory windows management directly in hardware where a RDMA write provide a memory semantic operation to write a virtually contiguous memory space on the remote node where the write wqe contains a gather list of local and virtual address of the remote memory space into which the local memory spaces are written (*paragraph 0048)* and a RDMA read provides a memory semantic operation to read a virtually contiguous memory space on a remote node where a memory

region references a previously registered set of virtually contiguous memory addresses defined by a virtual address and length *(paragraph 0046)*.

It would have been obvious to modify the system of Chung et al. by adding Beukema et al. infiniband memory windows management directly in hardware. A person of ordinary skill in the art at the time of applicant's invention would have been motivated to make the modification because it would allow better control access to memory areas within a computer *(paragraph 0008-0009)*.

In regard to claim 2, Chung et al. teach the system of Claim 1, further comprising:

a persistent memory manager *(Checkpoint Server, fig. 1, 110, col. 4 lines 41-44)* configured to provide address context information to the network interface *(pathname location of each copy of the application module on the host computer, fig. 2, 200)*.

Chung et al. does not teach the system comprising a persistent memory manager to keep the meta-data on the persistent memory unit consistent with the data stored on the persistent memory unit.

Beukema et al. teach of a memory region being an area of memory for which the translated physical addresses and access rights have been registered with the HCA *(paragraph 0005)*.

Refer to claim 1 for motivational statement.

In regard to claim 3, Chung et al. teach the system of Claim 1, wherein the persistent memory unit *(Checkpoint Server, fig. 1, 110, col. 4 lines 41-44)* is configured to provide read

access to the checkpoint data to another processor (*Checkpoint server transmit last stored state to new primary application module, fig. 1, 110, H1-6, col. 4 lines 41-48)*, and the backup process is executed by the other processor (*backup for A application on multiple host computer, fig. 2, H2 and H3)*.

Chung et al. does not teach a remote direct memory read access.

Beukema et al. teach a RDMA read provides a memory semantic operation to read a virtually contiguous memory space on a remote node where a memory region references a previously registered set of virtually contiguous memory addresses defined by a virtual address and length (*paragraph 0046).*

Refer to claim 1 for motivational statement.

In regard to claim 4, Chung et al. teach the system of Claim 1, wherein the persistent memory unit (*Checkpoint Server, fig. 1, 110, col. 4 lines 41-44)* provides the checkpoint data by the backup process after the primary process fails (*Checkpoint server transmit last stored state to new primary application module, fig. 1, 110, H1-6, col. 4 lines 41-48).*

Chung et al. does not teach remote direct memory reads.

Beukema et al. teach a RDMA read provides a memory semantic operation to read a virtually contiguous memory space on a remote node where a memory region references a previously registered set of virtually contiguous memory addresses defined by a virtual address and length (*paragraph 0046).*

Refer to claim 1 for motivational statement.

In regard to claim 5, Chung et al. teach the system of Claim 1, wherein the persistent memory unit (*Checkpoint Server, fig. 1, 110, col. 4 lines 41-44)* is configured to store multiple sets of checkpoint data sent from the processor at successive time intervals (*checkpoint technique to periodically take snapshots of the running state in a stable storage media, col. 1 lines 49-58).*

Chung et al. does not teach remote direct memory writes.

Beukema et al. teach a RDMA write provide a memory semantic operation to write a virtually contiguous memory space on the remote node where the write wqe contains a gather list of local and virtual address of the remote memory space into which the local memory spaces are written (*paragraph 0048).*

Refer to claim 1 for motivational statement.

In regard to claim 7, Chung et al. teach the system of Claim 1, wherein the primary process (*primary for application A, fig. 2, 202)* provides the checkpoint data (*snapshot of the running state of the primary application, col. 1 lines 49-58)* to the persistent memory unit (*Checkpoint Server, fig. 1, 110, col. 4 lines 41-44)* independently from the backup process (*backup for application A, fig. 2, 203-205).*

In regard to claim 8, Chung et al. teach the system of Claim 1, wherein the persistent memory unit (*Checkpoint Server, fig. 1, 110, col. 4 lines 41-44)* is configured as part of a access-enabled system area network (*Checkpoint Server is connected to network, fig. 1, 110, 100).*

Chung et al. does not teach a remote direct memory access.

Beukema et al. teach of a RDMA read and write (*paragraph 0046 and paragraph 0048).*

Refer to claim 1 for motivational statement.

In regard to claim 9, Chung et al. does not explicitly teach the system of Claim 1, wherein the persistent memory unit is configured with address protection and translation tables to authenticate requests from remote processors, and to provide access information to authenticated remote processors.

Beukema et al. teach a bind remote access key where the key is part of each RDMA access and is used to validate that the remote process has permitted access to the buffer (*paragraph 0050)* and the L_key is used to access the memory region's PTE which defines the characteristic of the Memory region and references the Address Translation Table that defines the virtual-to-real address mappings for the Memory Region (*paragraph 0061).*

Refer to claim 1 for motivational statement.

In regard to claim 10, Chung et al. teach a method for recovering the operational state of a primary process, comprising:

writing checkpoint data regarding the operational state of the primary process (*periodically receives from each fault-protected application module running on the network the most current state of that application, col. 4 lines 41-44)* to the persistent memory unit (*Checkpoint Server, fig. 1, 110, col. 4 lines 41-44)*; and

reading the checkpoint data from the persistent memory unit.

Chung et al. does not teach a method comprising: mapping virtual addresses of a persistent memory unit to physical addresses of the persistent memory unit in a remote direct memory write; remote direct memory read data subsequent to storing access information for the data to the physical addresses of the data in the persistent memory unit when the primary process opens a memory region for the checkpoint data; and providing the access information to subsequent requestors of the checkpoint data.

Beukema et al. teach an Address Translation Table that defines the virtual-to-real address mappings for the Memory Region (*paragraph 0061),* a RDMA read provides a memory semantic operation to read a virtually contiguous memory space on a remote node where a memory region references a previously registered set of virtually contiguous memory addresses defined by a virtual address and length (*paragraph 0046),* and where Bind Memory window provide the HCA hardware with the information required to change the access rights of a Memory Window (*paragraph 0053-0054).*

Refer to claim 1 for motivational statement.

In regard to claim 11, Chung et al. teach the method of Claim 10, further comprising: providing context information regarding the addresses to the primary process and the backup process. *It is inherent that the network of fig. 1 with the plurality of host computer H1-H6 if connected in an Ethernet network would have their own IP address to distinct one host computer from another (col. 3 lines 60-67). Furthermore, the registration request from each failure-*

protected application module included a list of the host computers on which the application modules resided and where on each the executable program can be found (col. 4 lines 49-60).

In regard to claim 12, Chung et al. teach the method of Claim 10, further comprising: reading the checkpoint data by the backup process upon failure of the primary process (*Checkpoint server transmit last stored state to new primary application module, fig. 1, 110, H1-6, col. 4 lines 41-48).*

Chung et al. does not teach remote direct memory reads.

Beukema et al. teach a RDMA read provides a memory semantic operation to read a virtually contiguous memory space on a remote node where a memory region references a previously registered set of virtually contiguous memory addresses defined by a virtual address and length (*paragraph 0046).*

Refer to claim 1 for motivational statement.

In regard to claim 13, Chung et al. teach the method of Claim 10, further comprising: overwriting the checkpoint data with current checkpoint data (*when a failure of the primary application, the checkpoint data of the last stored state of the failed primary is supplied to the backup application module, col. 1 lines 49-57).*

In regard to claim 18, Chung et al. does not teach the method of Claim 10, further comprising: establishing a connection to a process requesting access to the checkpoint data; and binding the access information to the connection.

Beukema et al. teach a RDMA read provides a memory semantic operation to read a virtually contiguous memory space on a remote node where a memory region references a previously registered set of virtually contiguous memory addresses defined by a virtual address and length (*paragraph 0046).*

Refer to claim 1 for motivational statement.

In regard to claim 19, Chung et al. does not teach the method of Claim 10, further comprising: verifying authentication information from the subsequent requestors.

Beukema et al. teach the method of checking that the access rights specified for the Memory Region allow the access requested in the Bind (*paragraph 0058).*

Refer to claim 1 for motivational statement.

In regard to claim 21, Chung et al. teach a computer product, comprising: computer executable instructions embodied in a computer readable medium and operable to:

allow access to a persistent memory unit from a remote processor via a network (*last stored state is retrieved from the memory of Checkpoint Server connected to network, fig. 1, 110, 100, col. 4 lines 41-48)*;

store checkpoint data from a primary process (*periodically receives from each fault-protected application module running on the network the most current state of that application, col. 4 lines 41-44)*; and

allow access to the checkpoint data for use in a backup process (*last operating state provided to the backup, col. 4 lines 34-40).*

Chung et al. does not teach a computer product comprising: a remote direct memory access references a persistent memory virtual address; authenticate requests from remote processors, and provide access information to authenticated remote processors based on address protection and translation tables in the persistent memory unit; translate the virtual address to a physical address in the persistent memory;

Beukema et al. teach the method of RDMA read and write (*paragraph 0046 and paragraph 0048)* and checking that the access rights specified for the Memory Region allow the access requested in the Bind (*paragraph 0058)* where a bind remote access key where the key is part of each RDMA access and is used to validate that the remote process has permitted access to the buffer (*paragraph 0050)* and the L_key is used to access the memory region's PTE which defines the characteristic of the Memory region and references the Address Translation Table that defines the virtual-to-real address mappings for the Memory Region (*paragraph 0061).*

Refer to claim 1 for motivational statement.

In regard to claim 25, Chung et al. teach the system of Claim 21, wherein the persistent memory (*Checkpoint Server, fig. 1, 110, col. 4 lines 41-44)* is configured as part of an access-enabled system area network (*Checkpoint Server is connected to network, fig. 1, 110, 100).*

Chung et al. does not teach a remote direct memory access.

Beukema et al. teach of a RDMA read and write (*paragraph 0046 and paragraph 0048).*

Refer to claim 1 for motivational statement.

In regard to claim 26, Chung et al. teach an apparatus comprising:

means for communicatively coupling (*configuring a fail-over process, col. 1 lines 36-37)* a persistent memory unit (*Checkpoint Server, fig. 1, 110, col. 4 lines 41-44)* to a network (*network, fig. 1, 110, col. 4 lines 41-44)* that enables read and write access to a persistent memory unit (*A checkpoint Server connected to network periodically receives from each fault-protected application module running on the network, col. 4 lines 41-44)*;

means for receiving the checkpoint data for a primary process in the persistent memory unit via the network (*periodically receives from each fault-protected application module running on the network the most current state of that application, col. 4 lines 41-44)*; and

means for allowing a backup process to access the checkpoint data via the network (*last operating state of the failed application module must be provided to the backup application module, col. 4 lines 34-40).*

Chung et al. does not teach an apparatus comprising: means for direct read and write access; means for receiving access information for physical addresses of checkpoint data in the persistent memory from the persistent memory unit; means for mapping virtual addresses of the persistent memory unit to physical addresses of the persistent memory unit;

Beukema et al. teach of a RDMA read and write (*paragraph 0046 and paragraph 0048)* and a RDMA read provides a memory semantic operation to read a virtually contiguous memory space on a remote node where a memory region references a previously registered set of virtually contiguous memory addresses defined by a virtual

address and length (*paragraph 0046),* and where Bind Memory window provide the HCA hardware with the information required to change the access rights of a Memory Window (*paragraph 0053-0054),* and an Address Translation Table that defines the virtual-to-real address mappings for the Memory Region (*paragraph 0061),*

Refer to claim 1 for motivational statement.

In regard to claim 28, Chung et al. teach the apparatus of claim 26, further comprising: means for allowing the backup process to access the checkpoint data upon failure of the primary process (*Checkpoint server transmit last stored state to new primary application module, fig. 1, 110, H1-6, col. 4 lines 41-48).*

In regard to claim 32, Chung et al. teach a method for recording the operational state of a primary process, comprising: accessing checkpoint data regarding the operational state of the primary process in a persistent memory unit (*periodically receives from each fault-protected application module running on the network the most current state of that application, col. 4 lines 41-44).*

Chung et al. does not teach a method comprising: receiving access information for physical addresses of checkpoint data in the persistent memory from the persistent memory unit; and accessing via a remote direct memory access write.

Beukema et al. teach a RDMA write (*paragraph 0048)* and memory region references a previously registered set of virtually contiguous memory addresses defined by a virtual address and length (*paragraph 0046),* and where Bind Memory window

provide the HCA hardware with the information required to change the access rights of a Memory Window (*paragraph 0053-0054).*

Refer to claim 1 for motivational statement.

In regard to claim 33, Chung et al. teach the method of Claim 32, further comprising: overwriting the checkpoint data in the persistent memory unit with current checkpoint data (*when a failure of the primary application, the checkpoint data of the last stored state of the failed primary is supplied to the backup application module, col. 1 lines 49-57).*

Beukema et al. teach a RDMA write provide a memory semantic operation to write a virtually contiguous memory space on the remote node where the write wqe contains a gather list of local and virtual address of the remote memory space into which the local memory spaces are written (*paragraph 0048).*

Refer to claim 1 for motivational statement.

In regard to claim 35, Chung et al. teach a method for retrieving the operational state of a primary process, comprising: transmitting a read command via network to a remote persistent memory unit from a backup process for the primary process (*the last stored state of that failed application module is retrieved from the memory of Checkpoint Server and provided to the new primary application module for continued processing, col. 4 lines 44-48).*

Chung et al. does not teach a method comprising: a remote direct memory access read and receiving access information for physical addresses of checkpoint data in the persistent memory from the persistent memory unit.

Beukema et al. teach of a RDMA read (*paragraph 0046)* and memory region references a previously registered set of virtually contiguous memory addresses defined by a virtual address and length (*paragraph 0046),* and where Bind Memory window provide the HCA hardware with the information required to change the access rights of a Memory Window (*paragraph 0053-0054).*

Refer to claim 1 for motivational statement.

In regard to claim 36, Chung et al. teach the method of Claim 35, further comprising: periodically transmitting the read command to retrieve at least a portion of the checkpoint data for the backup process (*Checkpoint Server periodically receives from each fault-protected application module running on the network the most current state of that application, col. 4 lines 41-44).*

Chung et al. does not teach the method comprising a remote direct memory access read command.

Beukema et al. teach of a RDMA read and write (*paragraph 0046 and paragraph 0048).*

Refer to claim 1 for motivational statement.

In regard to claim 37, Chung et al. teach the method of Claim 35, further comprising: transmitting the read command to retrieve previously unread portions of the checkpoint data upon failure of the primary process (*Checkpoint server transmit last stored state to new primary application module, fig. 1, 110, H1-6, col. 4 lines 41-48).*

Chung et al. does not teach the method comprising a remote direct memory access read command.

Beukema et al. teach of a RDMA read and write (*paragraph 0046 and paragraph 0048).*

Refer to claim 1 for motivational statement.

****

7. Claims 6 and 24 are rejected under 35 U.S.C. 103(a) as being unpatentable over Chung et al. (US 6,195,760) in further view of Beukema et al. (US 2002/0124117) in further view of Wang (US 7,082,553).

In regard to claim 6, Chung et al. does not explicitly teach the system of claim 5, wherein remote direct memory reads.

Beukema et al. teach a RDMA read provides a memory semantic operation to read a virtually contiguous memory space on a remote node where a memory region references a previously registered set of virtually contiguous memory addresses defined by a virtual address and length (*paragraph 0046).*

Refer to claim 1 for motivational statement.

Chung et al. and Beukema et al. does not teach the system wherein the persistent memory unit provides the multiple sets of checkpoint data upon request by the backup process at one time.

Wang teaches the for providing reliability and availability in a distributed component object model object oriented system by implementing checkpoint request to store current checkpoint of each object in registry image of disk storage (*fig. 8, col. 9 lines 21-30).*

It would have been obvious to modify the method of Chung et al. and Beukema et al. by adding Wang distributed component object model object oriented system. A person of ordinary skill in the art at the time of applicant's invention would have been motivated to make the modification because it would help to quickly recover from a failure (*col. 9 lines 30-46).*

In regard to claim 24, Chung et al. and Beukema et al. does not explicitly teach the computer product of claim 21, further comprising: computer executable instructions operable to: allow the backup process to access the multiple sets of the checkpoint data at one time.

Wang teaches the for providing reliability and availability in a distributed component object model object oriented system by implementing checkpoint request to store current checkpoint of each object in registry image of disk storage (*fig. 8, col. 9 lines 21-30).*

Refer to claim 6 for motivational statement.

****

8. Claims 14-16, 29-31, 34 are rejected under 35 U.S.C. 103(a) as being unpatentable over Chung et al. (US 6,195,760) in further view of Beukema et al. (US 2002/0124117) in further view of St. Pierre et al. (US 6,141,773).

In regard to claim 14, Chung et al. and Beukema et al. does not teach the method of claim 10, further comprising: appending updated checkpoint data to at least one previous set of the checkpoint data.

St. Pierre et al. disclosed the method of backing up and restoring data in a computer storage system where differential backup is formed by the identified changed segments omitting at least on the segments that has not been changed (*col. 5 lines 30-63).* A differential bit file may be constructed including copies of only the changed data segments (*fig. 13, 111a-111d).* The differential bit file captures changes to a logical entity as contiguous (*col. 17 lines 28-38).*

It would have been obvious to modify the method of Chung et al. and Beukema et al. by adding St. Pierre et al. method of backing up data in a computer storage system (*col. 5 lines 30-63).* A person of ordinary skill in the art at the time of applicant's invention would have been motivated to make the modification because it would permits recovery from errors, including use of a mirror for data located at a remote facility that also permits recoveries from catastrophic failure (*col. 5 lines 25-28).*

In regard to claim 15, Chung et al. teach the method of claim 10, further comprising: clearing the portion of the multiple sets of checkpoint data (*periodically receives state updates from the primary application module, col. 1 lines 64-67).*

In regard to claim 16, Chung et al. teach the method of Claim 14, further comprising:

allowing the backup process to read previously unread portions of the checkpoint data upon failure of the primary process (*Checkpoint server transmit last stored state to new primary application module, fig. 1, 110, H1-6, col. 4 lines 41-48)*; and

resuming functions performed by the primary process with the back process (*checkpoint data of the last stored state of the failed primary application module is supplied to the backup application module, col. 1 lines 52-58).*

Chung et al. does not teach remote direct memory reads.

Beukema et al. teach a RDMA read provides a memory semantic operation to read a virtually contiguous memory space on a remote node where a memory region references a previously registered set of virtually contiguous memory addresses defined by a virtual address and length (*paragraph 0046).*

Refer to claim 1 for motivational statement.

In regard to claim 29, Chung et al. teach the apparatus of claim 26, further comprising: means for overwriting the checkpoint data with current checkpoint data (*when a failure of the primary application, the checkpoint data of the last stored state of the failed primary is supplied to the backup application module, col. 1 lines 49-57).*

Chung et al. and Beukema et al. does not explicitly teach the means for creating multiple sets of checkpoint data by appending updated checkpoint data to at least one previous set of the checkpoint data;

St. Pierre et al. disclosed the method of backing up and restoring data in a computer storage system where differential backup is formed by the identified changed segments omitting at least on the segments that has not been changed (*col. 5 lines 30-63).* A differential bit file may be constructed including copies of only the changed data segments (*fig. 13, 111a-111d).* The differential bit file captures changes to a logical entity as contiguous (*col. 17 lines 28-38).*

Refer to claim 14 for motivational statement.

In regard to claim 30, Chung et al. disclosed the apparatus of claim 29, further comprising: means for periodically supplying (*Checkpoint server periodically receives from each fault-protected application module running on the network, fig. 1, col. 4 lines 41-44)* at least a portion of the multiple sets of checkpoint data (*snapshot of the running state of the primary application, col. 1 lines 49-58)* in the backup process (*Replica-Manager stores information necessary to effect recovery of an entire host computer running several different application modules, fig. 2, 200, col. 5 lines 21-30).*

In regard to claim 31, Chung et al. disclosed the apparatus of claim 30, further comprising: means for providing previously unread portions of the checkpoint data to the backup process upon failure of the primary process (*upon failure detection of an application module, the*

last stored state of the failed application is retrieved form the memory of Checkpoint Server, col. 4 lines 45-48).

In regard to claim 34, Chung et al. Beukema et al. does not explicitly teach the method of claim 32, further comprising: appending updated checkpoint data to a previous set of the checkpoint data via a direct memory access write command.

St. Pierre et al. disclosed the method of backing up and restoring data in a computer storage system where differential backup is formed by the identified changed segments omitting at least on the segments that has not been changed (*col. 5 lines 30-63).* A differential bit file may be constructed including copies of only the changed data segments (*fig. 13, 111a-111d).* The differential bit file captures changes to a logical entity as contiguous (*col. 17 lines 28-38).*

Refer to claim 14 for motivational statement.

****

9. Claim 20 is rejected under 35 U.S.C. 103(a) as being unpatentable over Chung et al. (US 6,195,760) in further view of Beukema et al. (US 2002/0124117) in further view of Ho et al. (US 2002/0073325).

In regard to claim 20, Chung et al. disclosed the method of claim 10, further comprising:

a persistent memory manager (*Checkpoint Server, fig. 1, 110, col. 4 lines 41-44)* during address protection and translation tables (*table, fig. 2, 200, col. 7 lines 1-5)* on the persistent memory unit (*Checkpoint Server, fig. 1, 110, col. 4 lines 41-44).*

Chung et al. and Beukema et al. does not explicitly teach the method of authenticating a persistent memory manager during initialization.

Ho et al. teach the method of authenticating software licenses by implementing a persistent storage medium comprising a signature authentication program in the software protection program (*paragraph 0023).*

It would have been obvious to modify the system of Chung et al. and Beukema et al. by adding Ho et al. method of authenticating software licenses. A person of ordinary skill in the art at the time of applicant's invention would have been motivated to make the modification because it would provide protection and allow legitimate backup copies (*paragraph 0011).*

****

10. Claims 22-23, 27 are rejected under 35 U.S.C. 103(a) as being unpatentable over Chung et al. (US 6,195,760) in further view of Beukema et al. (US 2002/0124117) in further view of Stiffer et al. (US 6,622,263).

In regard to claim 22, Chung et al. and Beukema et al. does not explicitly teach the computer product of claim 21, further comprising: computer executable instructions operable to: allow the processor to access address context information.

Stiffer et al. teach the method of checkpointing and fault recover software runs on standard platforms to detect observable malfunctions such as out-of-range address or exceeding a allocated range defined for a given data structure (*col. 4 lines 24-36).*

It would have been obvious to modify the system of Chung et al. and Beukema et al. by adding Stiffler et al. apparatus for achieving system-directed checkpointing without specialized hardware assistance. A person of ordinary skill in the art at the time of applicant's invention would have been motivated to make the modification because it would establishing and recording a consistent system state from which all running applications can be safely resumed following a fault (*col. 1 lines 13-17).*

In regard to claim 23, Chung et al. teach the computer product of claim 21, further comprising: computer executable instructions operable to: Store multiple updates to the checkpoint data sent at successive time intervals (*checkpoint technique to periodically take snapshots of the running state in a stable storage media, col. 1 lines 49-58).*

In regard to claim 27, Chung et al. and Beukema et al. does not teach the apparatus of Claim 26, further comprising: means for allowing the primary process and the backup process to access context information regarding the addresses.

Stiffer et al. teach the method of checkpointing and fault recover software runs on standard platforms to detect observable malfunctions such as out-of-range address or exceeding a allocated range defined for a given data structure (*col. 4 lines 24-36).*

Refer to claim 21 for motivational statement.

Conclusion

The prior art made of record and not relied upon is considered pertinent to applicant's disclosure. See PTO 892.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Loan Truong whose telephone number is (571) 272-2572. The examiner can normally be reached on M-F from 10am-6pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Scott Baderman can be reached on (571) 272-3644. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Loan Truong
Patent Examiner
AU 2114

/Scott T Baderman/
Supervisory Patent Examiner, Art Unit 2114